

Boltzmann Equation for WIMP Dark Matter

Jacinto Paulo Neto

June 2024

The Boltzmann equation for WIMP Dark matter is given by

$$\frac{dY_\chi(x)}{dx} = -\frac{s(x)\langle\sigma v\rangle}{xH(x)} [Y_\chi^2(x) - Y_\chi^{\text{eq}^2}(x)] \quad (1)$$

where $x = m_\chi/T$, the equilibrium comoving number density

$$Y_\chi^{\text{eq}}(x) = \frac{45}{4\pi^4} \frac{g_\chi}{g_{\star s}} x^2 K_2(x), \quad (2)$$

with g_χ corresponding to the dark matter degrees of freedom and $K_2(x)$ is the modified Bessel function of 2nd-kind. Moreover, the entropy density

$$s(x) = \frac{2\pi^2}{45} g_{\star s} m_\chi^3 x^{-3}, \quad (3)$$

and the Hubble rate

$$H(x) = \frac{\pi}{3} \sqrt{\frac{g_\star}{10}} \frac{m_\chi^2}{M_{\text{P}} x^2}, \quad (4)$$

with $M_{\text{P}} = 2 \times 10^{18}$ GeV. For the dark matter freeze-out $g_\star = g_{\star s} = 106.8$. On the next page, we exhibit a Python code to solve it numerically. The result must be equal or similar to Fig. 1.

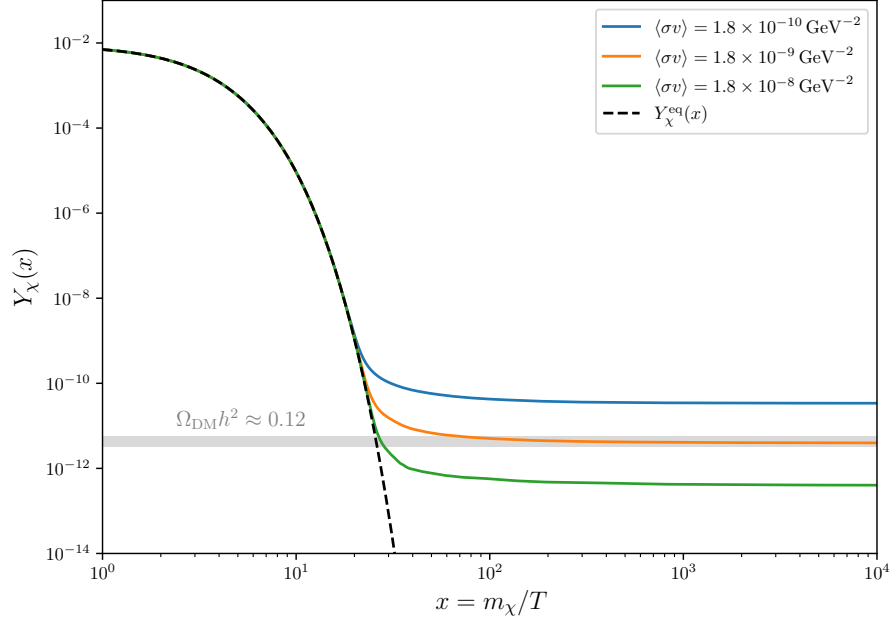


Figure 1: Numerical solution of the Boltzmann equation for WIMP Dark Matter.

```

1 import numpy as np
2 from scipy.integrate import solve_ivp
3 from scipy.special import kn
4 import matplotlib.pyplot as plt
5 from matplotlib import rc
6 rc('font', **{'family': 'serif', 'serif': ['Computer Modern']})
7 rc('text', usetex=True)
8 rc('text.latex', preamble=r'\usepackage{amsmath}')
9
10 # Functions
11 def s(x):
12     return (2 * np.pi**2 / 45) * gstar * mchi**3 * x**-3
13
14 def H(x):
15     return np.sqrt(np.pi**2 * gstar / 90) * mchi**2 / (Mp * x**2)
16
17 def Yeq(x):
18     return (45 / (4 * np.pi**4)) * (gchi / gstar) * x**2 * kn(2, x)
19
20 # Boltzmann equation
21 def boltzmann_eq(x, Y, sigmav):
22     Yeqx = Yeq(x)
23     return -s(x) * sigmav / (x * H(x)) * (Y**2 - Yeqx**2)
24
25 Markdown:  $\Omega_{\text{DM}} h^2 = 2.74385 \times 10^8 \frac{m_{\chi}}{\text{GeV}}$ 

```

```

26         \textrm{DM}}\}\{\textrm{GeV}} \quad Y_{\textrm{DM}}\}$
27 def YDMPlanck(mchi): # DM in GeV
28     return (0.12/2.74385)*(1e-8/mchi)
29
30 YDMPlanck(100)
31
32 # Constants
33 mchi = 100.0 # GeV
34 Mp = 2.435e18 # GeV
35 gchi = 4; gstar = 106.8
36
37 # Annihilation cross-sections
38 sigmav_values = [1.8e-10, 1.8e-9, 1.8e-8] # GeV^{-2}
39
40 # Initial conditions and integration range
41 xinit = 1e-2; xend = 1e4; Y0 = Yeq(xinit)
42
43 # Solving the differential equation for each cross-section
44 xvalues = np.logspace(np.log10(xinit), np.log10(xend), 1000)
45 solutions = {}
46
47 for sigmav in sigmav_values:
48     sol = solve_ivp(boltzmann_eq, [xinit, xend], [Y0], args=(sigmav
49         ,), dense_output=True, method='BDF', atol=1e-12, rtol=1e
50         -12)
51     Yvalues = sol.sol(xvalues)[0]
52     solutions[sigmav] = Yvalues
53
54 # Plotting the solution
55 fig = plt.figure()
56 fig, ax = plt.subplots(figsize=(7,5), tight_layout=True);
57
58 for sigmav in sigmav_values:
59     if(sigmav == 1.8e-10):
60         plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle = 10^{-10} \, \, \, \{\rm GeV\}^{-2}$')
61     if(sigmav == 1.8e-9):
62         plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle = 10^{-9} \, \, \, \{\rm GeV\}^{-2}$')
63     if(sigmav == 1.8e-8):
64         plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle = 10^{-8} \, \, \, \{\rm GeV\}^{-2}$')
65
66 plt.plot(xvalues, Yeq(xvalues), label=r'$Y_{\chi}(\rm eq)(x)$',
67     linestyle='--', color='black')
68 #plt.axhline(y=YDMPlanck(100), color='r', linestyle='--')
69 fig.text(0.2, 0.32, r"$\Omega_{\textrm{DM}} h^2 \approx 0.12$",
70     fontsize=12, color='gray')
71 plt.axhspan(YDMPlanck(100)+1.2e-12, YDMPlanck(100)-1.2e-12,
72     facecolor='gray', alpha = 0.3)
73 plt.xscale('log')
74 plt.yscale('log')
75 plt.xlabel(r'$x = m_{\chi} / T$', fontsize=14)
76 plt.ylabel(r'$Y_{\chi}(x)$', fontsize=14)
77 plt.xlim(1,1e4)
78 plt.ylim(1e-14,1e-1)

```

```
74 plt.legend()  
75 plt.savefig('WIMPDM-BEQ.pdf')  
76 plt.show()
```

Listing 1: Solution of the Boltzmann Equation for WIMP Dark Matter